

# Manufactured Solution for 2D transient Euler equations using Maple\*

Kémelli C. Estacio-Hiroms

August 19, 2010

## Abstract

This document describes the usage of the Method of Manufactured Solutions (MMS) for Code Verification of Navier–Stokes solvers for two-dimensional, unsteady flows. The manufactured solutions chosen for the flow variables are time-dependent, smooth and have non-trivial derivatives, so they may exercise all the terms of the governing equations without imposing special coordinate systems or particular boundary conditions. By the application of the MMS, a related set of governing equations is generated. The analytical solutions of the MMS-modified governing equations are, in turn, the manufactured solutions chosen a priori. Thus, the modified set of equations generated by the MMS may be discretized and solved numerically, convergence studies may be conducted, and the code may be verified.

## 1 2D Navier–Stokes Equations

The 2D Navier-Stokes equations in conservation form are:

$$\frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \quad (1)$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p - \tau_{xx})}{\partial x} + \frac{\partial(\rho uv - \tau_{xy})}{\partial y} = 0 \quad (2)$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu - \tau_{yx})}{\partial x} + \frac{\partial(\rho v^2 + p - \tau_{yy})}{\partial y} = 0 \quad (3)$$

$$\frac{\partial(\rho e_t)}{\partial t} + \frac{\partial(\rho u e_t + pu - u\tau_{xx} - v\tau_{xy} + q_x)}{\partial x} + \frac{\partial(\rho v e_t + pv - u\tau_{yx} - v\tau_{yy} + q_y)}{\partial y} = 0 \quad (4)$$

where the Equation (1) is the unsteady term (mass conservation), Equations (2) and (3) are the nonlinear convection term in the  $x$  and  $y$  direction (momentum), and Equation (4) is the total energy. Notice that Equations (2)–(4) include viscous effects.

For a calorically perfect gas, the Navier-Stokes equations are closed with two auxiliary relations for energy:

$$e = \frac{1}{\gamma - 1} RT \quad \text{and} \quad e_t = e + \frac{u^2 + v^2}{2}, \quad (5)$$

where  $\gamma$  is the ratio of specific heats, and with the ideal gas equation of state:

$$p = \rho RT. \quad (6)$$

The shear stress tensor is:

$$\begin{aligned} \tau_{xx} &= \frac{2}{3}\mu \left( 2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right), & \tau_{xy} &= \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right), \\ \tau_{yy} &= \frac{2}{3}\mu \left( 2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right), & \tau_{yx} &= \tau_{xy}, \end{aligned} \quad (7)$$

---

\*Work based on Salari and Knupp (2000).

where  $\mu$  is the absolute viscosity. The heat flux vector is given by:

$$q_x = -k \frac{\partial T}{\partial x} \quad \text{and} \quad q_y = -k \frac{\partial T}{\partial y}, \quad (8)$$

where  $k$  is the thermal conductivity, which can be determined by choosing the Prandtl number:

$$k = \frac{\gamma R \mu}{(\gamma - 1) \text{Pr}}.$$

## 2 Manufactured Solution

The Method of Manufactured Solutions (MMS) provides a general procedure for code accuracy verification (Roache, 2002; Bond et al., 2007). The MMS constructs a non-trivial but analytical solution for the flow variables; this manufactured solution usually does not satisfy the governing equations, since the choice is somewhat arbitrary. However, by passing the solution through the governing equations gives the production terms  $Q$ . A modified set of equations formed by adding these source terms to the right-hand-side of the original governing equations is forced to become a model for the constructed solution, i.e., the manufactured solutions chosen *a priori* are the analytical solutions of the MMS-modified equations.

Although the form of the manufactured solution is slightly arbitrary, it should be chosen to be smooth, infinitely differentiable and realizable (solutions should be avoided which have negative densities, pressures, temperatures, etc.) (Salari and Knupp, 2000; Roy et al., 2004). Solutions should also be chosen that are sufficiently general so as to exercise all terms in the governing equations. Examples of manufactured solutions and convergence studies for Euler and/or Navier–Stokes equations may be found in Salari and Knupp (2000); Roy et al. (2002, 2004); Bond et al. (2007); Orozco et al. (2010).

Salari and Knupp (2000) propose the general form of the primitive solution variables to be a function of sines or cosines:

$$\phi(x, y, t) = \phi_0 \left( f_s(x^2 + y^2 + \omega t) + \epsilon \right), \quad (9)$$

where  $\phi = \rho, u, v, p$  or  $e_t$ , and  $f_s(\cdot)$  functions denote either sine or cosine function. Note that  $\phi_0$ ,  $\omega$  and  $\epsilon$  are constants.

Therefore, a suitable set of time-dependent manufactured analytical solutions for for each one of the variables in Navier–Stokes equations is:

$$\begin{aligned} \rho(x, y, t) &= \rho_0 \left( \sin(x^2 + y^2 + \omega t) + \frac{3}{2} \right), \\ u(x, y, t) &= u_0 \left( \sin(x^2 + y^2 + \omega t) + \epsilon \right), \\ v(x, y, t) &= v_0 \left( \cos(x^2 + y^2 + \omega t) + \epsilon \right), \\ p(x, y, t) &= p_0 \left( \cos(x^2 + y^2 + \omega t) + 2 \right). \end{aligned} \quad (10)$$

Salari and Knupp (2000) present both the source for the 2D transient Navier–Stokes equations and a choice for the constants considering manufactured solutions for  $\rho$ ,  $u$ ,  $v$ , and  $e_t$ ; they also present source terms for the steady case with manufactured solutions for  $\rho$ ,  $u$ ,  $v$ , and  $p$ .

The source terms for the 2D transient Navier–Stokes equations (1)–(4) using time-dependent manufactured solutions for  $\rho$ ,  $u$ ,  $v$ , and  $p$ , described in Equations (10), are presented in the following sections.

## 3 Navier–Stokes mass conservation equation

The mass conservation equation written as an operator is:

$$L = \frac{\partial(\rho)}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} \quad (11)$$

Analytically differentiating Equation (10) for  $\rho$ ,  $u$  and  $v$  using operator  $L$  defined above gives the source term  $Q_\rho$ :

$$\begin{aligned} Q_\rho = & \rho_0 u_0 x \cos(\Phi) [4 \sin(\Phi) + 2\varepsilon + 3] + \\ & - \rho_0 v_0 y [4 \sin^2(\Phi) + 3 \sin(\Phi) - 2\varepsilon \cos(\Phi) - 2] + \\ & + \rho_0 \omega \cos(\Phi) \end{aligned} \quad (12)$$

where

$$\Phi = x^2 + y^2 + \omega t.$$

## 4 Navier–Stokes momentum equation

For the generation of the analytical source term  $Q_u$  for the  $x$  momentum equation, Equation (2) is written as an operator  $L$ :

$$L = \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p - \tau_{xx})}{\partial x} + \frac{\partial(\rho uv - \tau_{xy})}{\partial y}, \quad (13)$$

which, when operated in Equation (10), provides source term  $Q_u$ :

$$\begin{aligned} Q_u = & -2p_0 x \sin(\Phi) + \\ & + \rho_0 u_0^2 x [3 \sin(\Phi) \sin(2\Phi) + [4\varepsilon + 3] \sin(2\Phi) + 2\varepsilon[3 + \varepsilon] \cos(\Phi)] + \\ & - \rho_0 u_0 v_0 y [6 \sin^3(\Phi) + [4\varepsilon + 6] \sin^2(\Phi) + [3\varepsilon - 4] \sin(\Phi) - 2\varepsilon \sin(2\Phi) - \varepsilon[2\varepsilon + 3] \cos(\Phi) - 2\varepsilon - 3] + \\ & + \frac{\rho_0 u_0 \omega}{2} \cos(\Phi) [4 \sin(\Phi) + 2\varepsilon + 3] + \\ & + \frac{2\mu u_0}{3} [8x^2 + 6y^2] \sin(\Phi) - 7 \cos(\Phi) + \\ & + \frac{4\mu v_0 xy}{3} \cos(\Phi). \end{aligned} \quad (14)$$

Analogously, for the generation of the analytical source term  $Q_v$  for the  $y$  momentum equation, Equation (3) is written as an operator  $L$ :

$$L = \frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu - \tau_{yx})}{\partial x} + \frac{\partial(\rho v^2 + p - \tau_{yy})}{\partial y} \quad (15)$$

and then applied to Equation (10). It yields:

$$\begin{aligned} Q_v = & -2p_0 y \sin(\Phi) + \\ & - \rho_0 u_0 v_0 x [6 \sin^3(\Phi) + [4\varepsilon + 6] \sin^2(\Phi) + [3\varepsilon - 4] \sin(\Phi) - 2\varepsilon \sin(2\Phi) - \varepsilon[2\varepsilon + 3] \cos(\Phi) - 2\varepsilon - 3] + \\ & - \rho_0 v_0^2 y [8\varepsilon \sin^2(\Phi) + [3 \sin(2\Phi) + 6\varepsilon] \sin(\Phi) + 3 \sin(2\Phi) - [2\varepsilon^2 + 2] \cos(\Phi) - 4\varepsilon] + \\ & - \frac{\rho_0 v_0 \omega}{2} [4 \sin^2(\Phi) + 3 \sin(\Phi) - 2\varepsilon \cos(\Phi) - 2] + \\ & + \frac{4\mu u_0 xy}{3} \sin(\Phi) + \\ & + \frac{2\mu v_0}{3} [7 \sin(\Phi) + [6x^2 + 8y^2] \cos(\Phi)] \end{aligned} \quad (16)$$

where

$$\Phi = x^2 + y^2 + \omega t.$$

## 5 Navier–Stokes energy equation

The last component of Navier–Stokes equations is written as an operator:

$$L = \frac{\partial(\rho e_t)}{\partial t} + \frac{\partial(\rho u e_t + pu - u\tau_{xx} - v\tau_{xy} + q_x)}{\partial x} + \frac{\partial(\rho v e_t + pv - u\tau_{yx} - v\tau_{yy} + q_y)}{\partial y}. \quad (17)$$

Source term  $Q_e$  is obtained by operating  $L$  on Equation (10) together with the use of the auxiliary relations (5)–(6) for energy :

$$\begin{aligned} Q_e = & -2p_0 u_0 x \frac{\gamma}{\gamma-1} \left[ 2\sin^2(\Phi) + \varepsilon \sin(\Phi) - 2\cos(\Phi) - 1 \right] + \\ & -2p_0 v_0 y \frac{\gamma}{\gamma-1} \sin(\Phi) \left[ 2\cos(\Phi) + \varepsilon + 2 \right] + \\ & -\frac{p_0 \omega}{\gamma-1} \sin(\Phi) + \\ & + \frac{\rho_0 u_0^3 x}{2} \cos(\Phi) [\varepsilon + \sin(\Phi)]^2 [8\sin(\Phi) + 2\varepsilon + 9] + \\ & -\frac{\rho_0 v_0^3 y}{2} [\varepsilon + \cos(\Phi)]^2 \left[ 8\sin^2(\Phi) + 9\sin(\Phi) - 2\varepsilon \cos(\Phi) - 2 \right] + \\ & -\frac{\rho_0 u_0 v_0^2 x}{2} [\varepsilon + \cos(\Phi)] \left[ 8\sin^3(\Phi) + 3[2\varepsilon + 3]\sin^2(\Phi) + 2[3\varepsilon - 2]\sin(\Phi) - 2\varepsilon \sin(2\Phi) + \right. \\ & \quad \left. - [2\varepsilon + 3]\varepsilon \cos(\Phi) - 2\varepsilon - 3 \right] + \\ & -\frac{\rho_0 u_0^2 v_0 y}{2} [\varepsilon + \sin(\Phi)] \left[ 8\sin^3(\Phi) + [4\varepsilon + 9]\sin^2(\Phi) + 3[\varepsilon - 2]\sin(\Phi) - 3\varepsilon \sin(2\Phi) - 2[\varepsilon + 3]\varepsilon \cos(\Phi) + \right. \\ & \quad \left. - 2\varepsilon - 6 \right] + \\ & + \frac{\rho_0 u_0^2 \omega}{2} \cos(\Phi) [\varepsilon + \sin(\Phi)] [3\sin(\Phi) + \varepsilon + 3] + \\ & -\frac{\rho_0 v_0^2 \omega}{2} [\varepsilon + \cos(\Phi)] \left[ 3\sin^2(\Phi) + 3\sin(\Phi) - \varepsilon \cos(\Phi) - 1 \right] + \\ & + \frac{4\mu u_0 v_0 xy}{3} \left[ \varepsilon \sin(\Phi) + 2\sin(2\Phi) + \varepsilon \cos(\Phi) \right] + \\ & + \frac{2\mu u_0^2}{3} \left[ [16x^2 + 12y^2]\sin^2(\Phi) + [8\varepsilon x^2 + 6\varepsilon y^2]\sin(\Phi) - \frac{7}{2}\sin(2\Phi) - 7\varepsilon \cos(\Phi) - 8x^2 - 6y^2 \right] + \\ & -\frac{2\mu v_0^2}{3} \left[ [12x^2 + 16y^2]\sin^2(\Phi) - 7\varepsilon \sin(\Phi) - \frac{7}{2}\sin(2\Phi) + [-6\varepsilon x^2 - 8\varepsilon y^2]\cos(\Phi) - 6x^2 - 8y^2 \right] + \\ & + \frac{8kp_0}{R[2\sin(\Phi) + 3]^3 \rho_0} \left[ [8x^2 + 8y^2 + 6]\sin^2(\Phi) + [-12x^2 - 12y^2 + 13]\sin(\Phi) + [-3x^2 - 3y^2 + 4]\sin(2\Phi) + \right. \\ & \quad \left. + [x^2 + y^2 + 12]\cos(\Phi) - 16x^2 - 16y^2 + 6 \right] \end{aligned} \quad (18)$$

where

$$\Phi = x^2 + y^2 + \omega t.$$

## 6 Comments

Source terms  $Q_\rho$ ,  $Q_u$ ,  $Q_v$  and  $Q_e$  have been generated by replacing the analytical Expressions (10) into respective Equations (1) – (4), followed by the usage of Maple commands for collecting, sorting and factorizing the terms. Files containing C codes for the source terms have also been generated. They are: `NavierStokes_2d.transient_rho_code.C`, `NavierStokes_2d.transient_u_code.C`, `NavierStokes_2d.transient_v_code.C`, and `NavierStokes_2d.transient_e_code.C`.

An example of the automatically generated C file from the source term for the  $x$ -momentum equation is:

```
#include <math.h>

double SourceQ_u (double x, double y, double t, double rho_0, double p_0, double u_0,
                  double v_0, double omega, double epsilon, double mu)
{
    double Q_u;
    double Phi;
    Phi = x * x + y * y + omega * t;
    Q_u = (0.3e1 * sin(Phi) * sin(0.2e1 * Phi) + (0.4e1 * epsilon + 0.3e1) * sin(0.2e1 * Phi) +
          0.2e1 * epsilon * (0.3e1 + epsilon) * cos(Phi)) * rho_0 * u_0 * u_0 * x +
          (-0.6e1 * pow(sin(Phi), 0.3e1) + (-0.4e1 * epsilon - 0.6e1) * pow(sin(Phi), 0.2e1) +
          (-0.3e1 * epsilon + 0.4e1) * sin(Phi) + 0.2e1 * epsilon * sin(0.2e1 * Phi) +
          epsilon * (0.2e1 * epsilon + 0.3e1) * cos(Phi) + 0.2e1 * epsilon + 0.3e1) *
          rho_0 * u_0 * v_0 * y + (0.2e1 * sin(Phi) + epsilon + 0.3e1 / 0.2e1) *
          rho_0 * u_0 * omega * cos(Phi) + 0.4e1 / 0.3e1 * mu * v_0 * x * y * cos(Phi) -
          0.2e1 * p_0 * x * sin(Phi) + 0.7e1 * cos(Phi)) * u_0 -
          0.2e1 / 0.3e1 * mu * ((-0.8e1 * x * x - 0.6e1 * y * y) * sin(Phi));
    return(Q_u);
}
```

Maple's package for code generation also offers an option optimization of the code. Optimized versions of the code generated for each source code have also been created. An example of the code generated for  $Q_u$  using such option is:

```
#include <math.h>

double SourceQ_u (double x, double y, double t, double rho_0, double p_0, double u_0,
                  double v_0, double omega, double epsilon, double mu)
{
    double Phi;
    double t1;
    double t14;
    double t19;
    double t2;
    double t22;
    double t32;
    double t4;
    double t6;
    double t9;
    t1 = x * x;
    t2 = y * y;
    Phi = t1 + t2 + omega * t;
    t4 = sin(Phi);
    t6 = sin(0.2e1 * Phi);
    t9 = 0.4e1 * epsilon;
    t14 = cos(Phi);
    t19 = u_0 * u_0;
    t22 = t4 * t4;
    t32 = 0.2e1 * epsilon;
    return( (0.3e1 * t4 * t6 + (t9 + 0.3e1) * t6 + 0.2e1 * epsilon * (0.3e1 + epsilon) * t14) *
          rho_0 * t19 * x + (-0.6e1 * t22 * t4 + (-t9 - 0.6e1) * t22 + (-0.3e1 * epsilon + 0.4e1) *
          t4 + 0.2e1 * epsilon * t6 + epsilon * (t32 + 0.3e1) * t14 + t32 + 0.3e1) *
          rho_0 * u_0 * v_0 * y + (0.2e1 * sin(Phi) + epsilon + 0.3e1 / 0.2e1) *
          rho_0 * u_0 * omega * cos(Phi) + 0.4e1 / 0.3e1 * mu * v_0 * x * y * cos(Phi) -
          0.2e1 * p_0 * x * sin(Phi) + 0.7e1 * cos(Phi)) * u_0 -
          0.2e1 / 0.3e1 * mu * ((-0.8e1 * x * x - 0.6e1 * y * y) * sin(Phi));
}
```

```

rho_0 * u_0 * v_0 * y + (0.3e1 / 0.2e1 + 0.2e1 * t4 + epsilon) *
rho_0 * u_0 * omega * t14 + 0.4e1 / 0.3e1 * mu * v_0 * x * y * t14 - 0.2e1 * p_0 * x * t4 -
0.2e1 / 0.3e1 * mu * ((-0.8e1 * t1 - 0.6e1 * t2) * t4 + 0.7e1 * t14) * u_0 );
}

```

Finally, the gradients of the analytical solutions (9) have also been computed and their respective C codes are presented in `NavierStokes_2d.transient_manuf_solutions_grad.code.C`. Therefore,

$$\begin{aligned}
\nabla \rho &= \begin{bmatrix} 2\rho_0 x \cos(x^2 + y^2 + \omega t) \\ 2\rho_0 y \cos(x^2 + y^2 + \omega t) \end{bmatrix}, & \nabla p &= \begin{bmatrix} -2p_0 x \sin(x^2 + y^2 + \omega t) \\ -2p_0 y \sin(x^2 + y^2 + \omega t) \end{bmatrix}, \\
\nabla u &= \begin{bmatrix} 2u_0 x \cos(x^2 + y^2 + \omega t) \\ 2u_0 y \cos(x^2 + y^2 + \omega t) \end{bmatrix} & \text{and} & \nabla v &= \begin{bmatrix} -2v_0 x \sin(x^2 + y^2 + \omega t) \\ -2v_0 y \sin(x^2 + y^2 + \omega t) \end{bmatrix}
\end{aligned} \tag{19}$$

are written in C language as:

```

grad_rho_an[0] = 0.2e1 * rho_0 * cos(x * x + y * y + omega * t) * x;
grad_rho_an[1] = 0.2e1 * rho_0 * cos(x * x + y * y + omega * t) * y;
grad_p_an[0] = -0.2e1 * p_0 * sin(x * x + y * y + omega * t) * x;
grad_p_an[1] = -0.2e1 * p_0 * sin(x * x + y * y + omega * t) * y;
grad_u_an[0] = 0.2e1 * u_0 * cos(x * x + y * y + omega * t) * x;
grad_u_an[1] = 0.2e1 * u_0 * cos(x * x + y * y + omega * t) * y;
grad_v_an[0] = -0.2e1 * v_0 * sin(x * x + y * y + omega * t) * x;
grad_v_an[1] = -0.2e1 * v_0 * sin(x * x + y * y + omega * t) * y;

```

## References

- Bond, R., C. Ober, P. Knupp, and S. Bova (2007). Manufactured solution for computational fluid dynamics – boundary condition verification. *AIAA Journal* 45(9), 2224–2236.
- Orozco, C., D. Kizildag, A. Oliva, and C. D. Pérez-Segarra (2010). Verification of multidimensional and transient CFD solutions. *Numerical Heat Transfer, Part B: Fundamentals* 57(1), 46–73.
- Roache, P. (2002). Code verification by the method of manufactured solutions. *Journal of Fluids Engineering – Transactions of the ASME* 124(1), 4–10.
- Roy, C., C. Nelson, T. Smith, and C. Ober (2004). Verification of Euler/Navier–stokes codes using the method of manufactured solutions. *Int. J. Numer. Meth. Fluids* 44, 599–620.
- Roy, C., T. Smith, and C. Ober (2002). Verification of a compressible CFD code using the method of manufactured solutions. In *AIAA Fluid Dynamics Conference and Exhibit*, Number AIAA 2002-3110.
- Salari, K. and P. Knupp (2000). Code verification by the method of manufactured solutions. Technical Report SAND2000-1444, Sandia National Laboratories.